



Managing Software and Configurations

This chapter describes how to manage the ASA software and configurations and includes the following sections:

- [Upgrading the Software, page 42-1](#)
- [Managing Files, page 42-12](#)
- [Configuring the Images and Startup Configuration to Use, page 42-21](#)
- [Using the ROM Monitor to Load an Image, page 42-22](#)
- [Backing Up Configurations or Other Files, page 42-25](#)
- [Downgrading Your Software, page 42-34](#)
- [Configuring Auto Update, page 42-35](#)

Upgrading the Software

This section describes how to upgrade to the latest version and includes the following topics:

- [Upgrade Path and Migrations, page 42-1](#)
- [Viewing Your Current Version, page 42-3](#)
- [Downloading the Software from Cisco.com, page 42-3](#)
- [Upgrading a Standalone Unit, page 42-3](#)
- [Upgrading a Failover Pair or ASA Cluster, page 42-5](#)

Upgrade Path and Migrations

- If you are upgrading from a pre-8.3 release:
 - See the [Cisco ASA 5500 Migration Guide to Version 8.3 and Later](#) for important information about migrating your configuration.
 - You cannot upgrade directly to 9.0 or later. You must first upgrade to Version 8.3 or 8.4 for a successful migration.
- If you are upgrading from a pre-9.0 release, because of ACL migration, you cannot later perform a downgrade; be sure to back up your configuration file in case you want to downgrade. See the ACL migration section in the 9.0 release notes for more information.

- For a pre-9.1(2.8) version, to upgrade to 9.1(2.8) or 9.1(3) and later, you must be running one of the following versions:
 - 8.4(5) or later
 - 9.0(2) or later
 - 9.1(2)

However, if you are running any earlier versions, you cannot upgrade directly to 9.1(2.8) or 9.1(3) or later without *first* upgrading to one of the above versions. For example:

Pre-9.1(2.8) ASA Version	First Upgrade to:	Then Upgrade to:
8.2(1)	8.4(6)	9.1(2.8) or 9.1(3) or later
8.4(4)	8.4(6)	9.1(2.8) or 9.1(3) or later
9.0(1)	9.0(3)	9.1(2.8) or 9.1(3) or later
9.1(1)	9.1(2)	9.1(2.8) or 9.1(3) or later



- Software Version Requirements for Zero Downtime Upgrading:
 - The units in a failover configuration or ASA cluster should have the same major (first number) and minor (second number) software version. However, you do not need to maintain version parity on the units during the upgrade process; you can have different versions on the software running on each unit and still maintain failover support. To ensure long-term compatibility and stability, we recommend upgrading all units to the same version as soon as possible.

[Table 42-1](#) shows the supported scenarios for performing zero-downtime upgrades.

Table 42-1 Zero-Downtime Upgrade Support

Type of Upgrade	Support
Maintenance Release	<p>You can upgrade from any maintenance release to any other maintenance release within a minor release.</p> <p>For example, you can upgrade from 8.4(1) to 8.4(6) without first installing the maintenance releases in between.</p>

Table 42-1 Zero-Downtime Upgrade Support (continued)

Type of Upgrade	Support
Minor Release	<p>You can upgrade from a minor release to the next minor release. You cannot skip a minor release.</p> <p>For example, you can upgrade from 8.2 to 8.3. Upgrading from 8.2 directly to 8.4 is not supported for zero-downtime upgrades; you must first upgrade to 8.3. For models that are not supported on a minor release, you can skip the minor release; for example, for the ASA 5585-X, you can upgrade from 8.2 to 8.4 (the model is not supported on 8.3).</p> <hr/> <p> Note Zero-downtime upgrades are possible, even when feature configuration is migrated, for example, from 8.2 to 8.3.</p>
Major Release	<p>You can upgrade from the last minor release of the previous version to the next major release.</p> <p>For example, you can upgrade from 8.6 to 9.0, assuming that 8.6 is the last minor version in the 8.x release series for your model. Upgrading from 8.6 directly to 9.1 is not supported for zero-downtime upgrades; you must first upgrade to 9.0. For models that are not supported on a minor release, you can skip the minor release; for example, for the ASA 5585-X, you can upgrade from 8.4 to 9.0 (the model is not supported on 8.5 or 8.6).</p> <hr/> <p> Note Zero-downtime upgrades are possible, even when feature configuration is migrated, for example, from 8.4 to 9.0.</p>

Viewing Your Current Version

Use the **show version** command to verify the software version of your ASA.

Downloading the Software from Cisco.com

If you have a Cisco.com login, you can obtain the OS and ASDM images from the following website:

<http://www.cisco.com/go/asa-software>

This procedure assumes you put the images on a TFTP server, although other server types are supported.

Upgrading a Standalone Unit

This section describes how to install the ASDM and operating system (OS) images.

Detailed Steps

This procedure uses TFTP. For FTP or HTTP, see the **copy** command.

	Command	Purpose
Step 1	<pre>more system:running-config</pre> <p>Example: hostname# more system:running-config</p>	<p>(If there is a configuration migration) The output shows the configuration on the terminal so that you can back up your configuration. Copy the output from this command, then paste the configuration in to a text file.</p> <p>For other methods of backing up, see the configuration guide.</p>
Step 2	<pre>copy tftp://server[/path]/asa_image_name {disk0:/ disk1:/}[path/]asa_image_name</pre> <p>Example: hostname# copy tftp://10.1.1.1/asa911-smp-k8.bin disk0:/asa911-smp-k8.bin</p>	<p>Copies the ASA software to the active unit flash memory. For other methods than TFTP, see the copy command.</p>
Step 3	<pre>copy tftp://server[/path]/asdm_image_name {disk0:/ disk1:/}[path/]asdm_image_name</pre> <p>Example: hostname# copy tftp://10.1.1.1/asdm-711.bin disk0:/asdm-711.bin</p>	<p>Copies the ASDM image to the active unit flash memory.</p>
Step 4	<pre>configure terminal</pre> <p>Example: hostname(config)# configure terminal</p>	<p>If you are not already in global configuration mode, accesses global configuration mode.</p>
Step 5	<pre>show running-config boot system</pre> <p>Example: hostname(config)# show running-config boot system boot system disk0:/cdisk.bin boot system disk0:/asa841-smp-k8.bin</p>	<p>Shows the current boot images configured (up to 4). The ASA uses the images in the order listed; if the first image is unavailable, the next image is used, and so on. You cannot insert a new image URL at the top of the list; to specify the new image to be first, you must remove any existing entries, and enter the image URLs in the order desired, according to Step 6 and Step 7.</p>
Step 6	<pre>no boot system {disk0:/ disk1:/}[path/]asa_image_name</pre> <p>Example: hostname(config)# no boot system disk0:/cdisk.bin hostname(config)# no boot system disk0:/asa841-smp-k8.bin</p>	<p>Removes any existing boot image configurations so that you can enter the new boot image as your first choice.</p>
Step 7	<pre>boot system {disk0:/ disk1:/}[path/]asa_image_name</pre> <p>Example: hostname(config)# boot system disk0://asa911-smp-k8.bin</p>	<p>Sets the ASA image to boot (the one you just uploaded).</p> <p>Repeat this command for any backup images that you want to use in case this image is unavailable. For example, you can re-enter the images that you previously removed in Step 6.</p>

	Command	Purpose
Step 8	<pre>asdm image {disk0:/ disk1:/} [path/] asdm_image_name</pre> <p>Example: hostname(config)# asdm image disk0:/asdm-711.bin </p>	Sets the ASDM image to use (the one you just uploaded). You can only configure one ASDM image to use, so you do not need to first remove the existing configuration.
Step 9	<pre>write memory</pre> <p>Example: hostname(config)# write memory </p>	Saves the new settings to the startup configuration.
Step 10	<pre>reload</pre> <p>Example: hostname# reload </p>	Reloads the ASA.

Upgrading a Failover Pair or ASA Cluster

- [Upgrading an Active/Standby Failover Pair, page 42-5](#)
- [Upgrading an Active/Active Failover Pair, page 42-8](#)
- [Upgrading an ASA Cluster, page 42-10](#)

Upgrading an Active/Standby Failover Pair

To upgrade the Active/Standby failover pair, perform the following steps.

Requirements

Perform these steps on the active unit.

Detailed Steps

	Command	Purpose
Step 1	<pre>more system:running-config</pre> <p>Example: active# more system:running-config </p>	(If there is a configuration migration) The output shows the configuration on the terminal so that you can back up your configuration. Copy the output from this command, then paste the configuration in to a text file. For other methods of backing up, see the configuration guide.
Step 2	<pre>copy tftp://server[/path]/asa_image_name {disk0:/ disk1:/} [path/] asa_image_name</pre> <p>Example: active# copy tftp://10.1.1.1/asa911-smp-k8.bin disk0:/asa911-smp-k8.bin </p>	Copies the ASA software to the active unit flash memory. For other methods than TFTP, see the copy command.

	Command	Purpose
Step 3	<pre>failover exec mate copy /noconfirm tftp://server[/path]/filename {disk0:/ disk1:/}[path/] filename</pre> <p>Example:</p> <pre>active# failover exec mate copy /noconfirm tftp://10.1.1.1/asa911-smp-k8.bin disk0:/asa911-smp-k8.bin</pre>	Copies the software to the standby unit; be sure to specify the same path as for the active unit.
Step 4	<pre>copy tftp://server[/path]/asdm_image_name {disk0:/ disk1:/}[path/] asdm_image_name</pre> <p>Example:</p> <pre>active# copy tftp://10.1.1.1/asdm-711.bin disk0:/asdm-711.bin</pre>	Copies the ASDM image to the active unit flash memory.
Step 5	<pre>failover exec mate copy /noconfirm tftp://server[/path]/asdm_image_name {disk0:/ disk1:/}[path/] asdm_image_name</pre> <p>Example:</p> <pre>active# failover exec mate copy /noconfirm tftp://10.1.1.1/asdm-711.bin disk0:/asdm-711.bin</pre>	Copies the ASDM image to the standby unit; be sure to specify the same path as for the active unit.
Step 6	<pre>configure terminal</pre> <p>Example:</p> <pre>active(config)# configure terminal</pre>	If you are not already in global configuration mode, accesses global configuration mode.
Step 7	<pre>show running-config boot system</pre> <p>Example:</p> <pre>hostname(config)# show running-config boot system boot system disk0:/cdisk.bin boot system disk0:/asa841-smp-k8.bin</pre>	Shows the current boot images configured (up to 4). The ASA uses the images in the order listed; if the first image is unavailable, the next image is used, and so on. You cannot insert a new image URL at the top of the list; to specify the new image to be first, you must remove any existing entries, and enter the image URLs in the order desired, according to Step 8 and Step 9 .
Step 8	<pre>no boot system {disk0:/ disk1:/}[path/] asa_image_name</pre> <p>Example:</p> <pre>hostname(config)# no boot system disk0:/cdisk.bin hostname(config)# no boot system disk0:/asa841-smp-k8.bin</pre>	Removes any existing boot image configurations so that you can enter the new boot image as your first choice.
Step 9	<pre>boot system {disk0:/ disk1:/}[path/] asa_image_name</pre> <p>Example:</p> <pre>hostname(config)# boot system disk0://asa911-smp-k8.bin</pre>	Sets the ASA image to boot (the one you just uploaded). Repeat this command for any backup images that you want to use in case this image is unavailable. For example, you can re-enter the images that you previously removed in Step 8 .

	Command	Purpose
Step 10	<pre>asdm image {disk0:/ disk1:/} [path/] asdm_image_name</pre> <p>Example: hostname(config)# asdm image disk0:/asdm-711.bin</p>	Sets the ASDM image to use (the one you just uploaded). You can only configure one ASDM image to use, so you do not need to first remove the existing configuration.
Step 11	<pre>write memory</pre> <p>Example: active(config)# write memory</p>	Saves the new settings to the startup configuration.
Step 12	<pre>failover reload-standby</pre> <p>Example: active# failover reload-standby</p>	<p>Reloads the standby unit to boot the new image.</p> <p>Wait for the standby unit to finish loading. Use the show failover command to verify that the standby unit is in the Standby Ready state.</p>
Step 13	<pre>no failover active</pre> <p>Example: active# no failover active</p>	Forces the active unit to fail over to the standby unit.
Step 14	<pre>reload</pre> <p>Example: active# reload</p>	Reloads the former active unit (now the new standby unit). If you want to restore this unit to be active after it reloads, enter the failover active command.

Upgrading an Active/Active Failover Pair

To upgrade two units in an Active/Active failover configuration, perform the following steps.

Requirements

Perform these steps in the system execution space. Also perform these steps on the *primary* unit.

Detailed Steps

	Command	Purpose
Step 1	<pre>more system:running-config</pre> <p>Example: primary# more system:running-config </p>	<p>(If there is a configuration migration) The output shows the configuration on the terminal so that you can back up your configuration. Copy the output from this command, then paste the configuration in to a text file.</p> <p>For other methods of backing up, see the configuration guide.</p>
Step 2	<pre>copy tftp://server[/path]/asa_image_name {disk0:/ disk1:/}[path]/asa_image_name</pre> <p>Example: primary# copy tftp://10.1.1.1/asa911-smp-k8.bin disk0:/asa911-smp-k8.bin </p>	<p>Copies the ASA software to the primary unit flash memory. For other methods than TFTP, see the copy command.</p>
Step 3	<pre>failover exec mate copy /noconfirm tftp://server[/path]/filename {disk0:/ disk1:/}[path]/filename</pre> <p>Example: primary# failover exec mate copy /noconfirm tftp://10.1.1.1/asa911-smp-k8.bin disk0:/asa911-smp-k8.bin </p>	<p>Copies the software to the secondary unit; be sure to specify the same path as for the primary unit.</p>
Step 4	<pre>copy tftp://server[/path]/asdm_image_name {disk0:/ disk1:/}[path]/asdm_image_name</pre> <p>Example: primary# copy tftp://10.1.1.1/asdm-711.bin disk0:/asdm-711.bin </p>	<p>Copies the ASDM image to the primary unit flash memory.</p>
Step 5	<pre>failover exec mate copy /noconfirm tftp://server[/path]/asdm_image_name {disk0:/ disk1:/}[path]/asdm_image_name</pre> <p>Example: primary# failover exec mate copy /noconfirm tftp://10.1.1.1/asdm-711.bin disk0:/asdm-711.bin </p>	<p>Copies the ASDM image to the secondary unit; be sure to specify the same path as for the active unit.</p>

	Command	Purpose
Step 6	<pre>failover active group 1 failover active group 2</pre> <p>Example:</p> <pre>primary# failover active group 1 primary# failover active group 2</pre>	Makes both failover groups active on the primary unit.
Step 7	<pre>configure terminal</pre> <p>Example:</p> <pre>primary(config)# configure terminal</pre>	If you are not already in global configuration mode, accesses global configuration mode.
Step 8	<pre>show running-config boot system</pre> <p>Example:</p> <pre>hostname(config)# show running-config boot system boot system disk0:/cdisk.bin boot system disk0:/asa841-smp-k8.bin</pre>	Shows the current boot images configured (up to 4). The ASA uses the images in the order listed; if the first image is unavailable, the next image is used, and so on. You cannot insert a new image URL at the top of the list; to specify the new image to be first, you must remove any existing entries, and enter the image URLs in the order desired, according to Step 9 and Step 10 .
Step 9	<pre>no boot system {disk0:/ disk1:/}[path/]asa_image_name</pre> <p>Example:</p> <pre>hostname(config)# no boot system disk0:/cdisk.bin hostname(config)# no boot system disk0:/asa841-smp-k8.bin</pre>	Removes any existing boot image configurations so that you can enter the new boot image as your first choice.
Step 10	<pre>boot system {disk0:/ disk1:/}[path/]asa_image_name</pre> <p>Example:</p> <pre>hostname(config)# boot system disk0://asa911-smp-k8.bin</pre>	Sets the ASA image to boot (the one you just uploaded). Repeat this command for any backup images that you want to use in case this image is unavailable. For example, you can re-enter the images that you previously removed in Step 9 .
Step 11	<pre>asdm image {disk0:/ disk1:/}[path/]asdm_image_name</pre> <p>Example:</p> <pre>hostname(config)# asdm image disk0:/asdm-711.bin</pre>	Sets the ASDM image to use (the one you just uploaded). You can only configure one ASDM image to use, so you do not need to first remove the existing configuration.
Step 12	<pre>write memory</pre> <p>Example:</p> <pre>primary(config)# write memory</pre>	Saves the new settings to the startup configuration.
Step 13	<pre>failover reload-standby</pre> <p>Example:</p> <pre>primary# failover reload-standby</pre>	Reloads the secondary unit to boot the new image. Wait for the secondary unit to finish loading. Use the show failover command to verify that both failover groups are in the Standby Ready state.

	Command	Purpose
Step 14	<pre>no failover active group 1 no failover active group 2</pre> <p>Example:</p> <pre>primary# no failover active group 1 primary# no failover active group 2</pre>	Forces both failover groups to become active on the secondary unit.
Step 15	<pre>reload</pre> <p>Example:</p> <pre>primary# reload</pre>	Reloads the primary unit. If the failover groups are configured with the preempt command, they automatically become active on their designated unit after the preempt delay has passed. If the failover groups are not configured with the preempt command, you can return them to active status on their designated units using the failover active group command.

Upgrading an ASA Cluster

To upgrade all units in an ASA cluster, perform the following steps on the master unit. For multiple context mode, perform these steps in the system execution space.

Detailed Steps

	Command	Purpose
Step 1	<pre>more system:running-config</pre> <p>Example:</p> <pre>master# more system:running-config</pre>	<p>(If there is a configuration migration) Backs up your configuration file. Copy the output from this command, then paste the configuration in to a text file.</p> <p>For other methods of backing up, see the configuration guide.</p>
Step 2	<pre>cluster exec copy /noconfirm tftp://server[/path]/asa_image_name {disk0:/ disk1:/}[path/]asa_image_name</pre> <p>Example:</p> <pre>master# cluster exec copy /noconfirm tftp://10.1.1.1/asa911-smp-k8.bin disk0:/asa911-smp-k8.bin</pre>	Copies the ASA software to all units in the cluster. For other methods than TFTP, see the copy command.
Step 3	<pre>cluster exec copy /noconfirm tftp://server[/path]/asdm_image_name {disk0:/ disk1:/}[path/]asdm_image_name</pre> <p>Example:</p> <pre>master# cluster exec copy /noconfirm tftp://10.1.1.1/asdm-711.bin disk0:/asdm-711.bin</pre>	Copies the ASDM image to all units in the cluster.
Step 4	<pre>configure terminal</pre> <p>Example:</p> <pre>master(config)# configure terminal</pre>	If you are not already in global configuration mode, accesses global configuration mode.

	Command	Purpose
Step 5	<p>show running-config boot system</p> <p>Example: hostname(config)# show running-config boot system boot system disk0:/cdisk.bin boot system disk0:/asa841-smp-k8.bin</p>	Shows the current boot images configured (up to 4). The ASA uses the images in the order listed; if the first image is unavailable, the next image is used, and so on. You cannot insert a new image URL at the top of the list; to specify the new image to be first, you must remove any existing entries, and enter the image URLs in the order desired, according to Step 6 and Step 7 .
Step 6	<p>no boot system {disk0:/ disk1:/} [path/]asa_image_name</p> <p>Example: hostname(config)# no boot system disk0:/cdisk.bin hostname(config)# no boot system disk0:/asa841-smp-k8.bin</p>	Removes any existing boot image configurations so that you can enter the new boot image as your first choice.
Step 7	<p>boot system {disk0:/ disk1:/} [path/]asa_image_name</p> <p>Example: hostname(config)# boot system disk0://asa911-smp-k8.bin</p>	Sets the ASA image to boot (the one you just uploaded). Repeat this command for any backup images that you want to use in case this image is unavailable. For example, you can re-enter the images that you previously removed in Step 6 .
Step 8	<p>asdm image {disk0:/ disk1:/} [path/]asdm_image_name</p> <p>Example: hostname(config)# asdm image disk0:/asdm-711.bin</p>	Sets the ASDM image to use (the one you just uploaded). You can only configure one ASDM image to use, so you do not need to first remove the existing configuration.
Step 9	<p>write memory</p> <p>Example: master(config)# write memory</p>	Saves the new settings to the startup configuration.
Step 10	<p>cluster exec unit slave-unit reload noconfirm</p> <p>Example: master# cluster exec unit unit2 reload noconfirm</p>	<p>Reloads each slave unit when you repeat this command for each unit name. To avoid connection loss and allow traffic to stabilize, wait for each unit to come back up (approximately 5 minutes) before reloading the next unit.</p> <p>To view member names, enter cluster exec unit ?, or enter the show cluster info command.</p>
Step 11	<p>no enable</p> <p>Example: master(config)# no enable</p>	<p>Disables clustering on the master unit. Wait for 5 minutes for a new master to be selected and traffic to stabilize.</p> <p>Do not enter write memory; when the master unit reloads, you want clustering to be enabled on it.</p>
Step 12	<p>reload noconfirm</p> <p>Example: master# reload noconfirm</p>	Reloads the master unit. A new election takes place for a new master unit. When the former master unit rejoins the cluster, it will be a slave.

Managing Files

- [Viewing Files in Flash Memory, page 42-12](#)
- [Deleting Files from Flash Memory, page 42-12](#)
- [Erasing the Flash File System, page 42-13](#)
- [Configuring File Access, page 42-13](#)
- [Copying a File to the ASA, page 42-17](#)
- [Copying a File to the Startup or Running Configuration, page 42-19](#)

Viewing Files in Flash Memory

You can view files in flash memory and see information about files as follows:

- To view files in flash memory, enter the following command:

```
ciscoasa# dir [disk0: | disk1:]
```

Enter **disk0:** for the internal flash memory. The **disk1:** keyword represents the external flash memory. The internal flash memory is the default.

For example:

```
hostname# dir
```

```
Directory of disk0:/
500  -rw-  4958208    22:56:20 Nov 29 2004  cdisk.bin
2513 -rw-   4634      19:32:48 Sep 17 2004  first-backup
2788 -rw-   21601    20:51:46 Nov 23 2004  backup.cfg
2927 -rw-  8670632    20:42:48 Dec 08 2004  asdmfile.bin
```

- To view extended information about a specific file, enter the following command:

```
hostname# show file information [path:] filename
```

The default path is the root directory of the internal flash memory (disk0:/).

For example:

```
hostname# show file information cdisk.bin
```

```
disk0:/cdisk.bin:
  type is image (XXX) []
  file size is 4976640 bytes version 7.0(1)
```

The file size listed is for example only.

Deleting Files from Flash Memory

You can remove files from flash memory that you no longer need. To delete a file from flash memory, enter the following command:

```
hostname# delete disk0: filename
```

By default, the file is deleted from the current working directory if you do not specify a path. You may use wildcards when deleting files. You are prompted with the filename to delete, and then you must confirm the deletion.

Erasing the Flash File System

To erase the flash file system, perform the following steps:

-
- Step 1** Connect to the ASA console port according to the instructions in the [“Accessing the ASA Services Module Command-Line Interface”](#) section on page 3-2 or the [“Accessing the Appliance Command-Line Interface”](#) section on page 3-1.
 - Step 2** Power off the ASA, then power it on.
 - Step 3** During startup, press the **Escape** key when you are prompted to enter ROMMON mode.
 - Step 4** Enter the **erase** command, which overwrites all files and erases the file system, including hidden system files.

```
rommon #1> erase [disk0: | disk1: | flash:]
```

Configuring File Access

- [Configuring the FTP Client Mode, page 42-13](#)
- [Configuring the ASA as a Secure Copy Server, page 42-14](#)
- [Customizing the ASA Secure Copy Client, page 42-14](#)
- [Configuring the ASA TFTP Client Path, page 42-16](#)

Configuring the FTP Client Mode

The ASA can use FTP to upload or download image files or configuration files to or from an FTP server. In passive FTP, the client initiates both the control connection and the data connection. The server, which is the recipient of the data connection in passive mode, responds with the port number to which it is listening for the specific connection.

Detailed Steps

Command	Purpose
<code>ftp mode passive</code>	Sets the FTP mode to passive.
Example: <code>ciscoasa(config)# ftp mode passive</code>	

Configuring the ASA as a Secure Copy Server

You can enable the secure copy (SCP) server on the ASA. Only clients that are allowed to access the ASA using SSH can establish a secure copy connection.

Restrictions

- The server does not have directory support. The lack of directory support limits remote client access to the ASA internal files.
- The server does not support banners.
- The server does not support wildcards.

Prerequisites

- Enable SSH on the ASA according to the [“Configuring ASA Access for ASDM, Telnet, or SSH” section on page 41-1](#).
- The ASA license must have the strong encryption (3DES/AES) license to support SSH Version 2 connections.

Detailed Steps

Command	Purpose
<code>ssh scopy enable</code>	Enables the SCP server.
Example: <code>ciscoasa(config)# ssh scopy enable</code>	

Example

From a client on the external host, perform an SCP file transfer. For example, in Linux enter the following command:

```
scp -v -pw password source_filename username@asa_address:{disk0|disk1}:/dest_filename
```

The `-v` is for verbose, and if `-pw` is not specified, you will be prompted for a password.

Customizing the ASA Secure Copy Client

You can copy files to and from the ASA using the on-board SCP client (see the [“Copying a File to the ASA” section on page 42-17](#)). This section lets you customize the SCP client operation.

Prerequisites

For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the `changeto system` command.

Detailed Steps

	Command	Purpose
<p>Step 1</p> <pre>[no] ssh stricthostkeycheck</pre> <p>Example:</p> <pre>ciscoasa# ssh stricthostkeycheck ciscoasa# copy x scp://cisco@10.86.95.9/x The authenticity of host '10.86.95.9 (10.86.95.9)' can't be established. RSA key fingerprint is dc:2e:b3:e4:e1:b7:21:eb:24:e9:37:81:cf:bb: c3:2a. Are you sure you want to continue connecting (yes/no)? yes Warning: Permanently added '10.86.95.9' (RSA) to the list of known hosts. Source filename [x]? Address or name of remote host [10.86.95.9]? Destination username [cisco]? Destination password []? cisco123 Destination filename [x]?</pre>	<p>Enables or disables SSH host key checking. By default, this option is enabled. When this option is enabled, you are prompted to accept or reject the host key if it is not already stored on the ASA. When this option is disabled, the ASA accepts the host key automatically if it was not stored before.</p>	
<p>Step 2</p> <pre>ssh pubkey-chain [no] server ip_address {key-string key_string exit key-hash {md5 sha256} fingerprint}</pre> <p>Example:</p> <pre>ciscoasa(config)# ssh pubkey-chain ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9 ciscoasa(config-ssh-pubkey-server)# key-string Enter the base 64 encoded RSA public key. End with the word "exit" on a line by itself ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:46:41: 63:87 ciscoasa(config-ssh-pubkey-server-string)# exit ciscoasa(config-ssh-pubkey-server)# show running-config ssh pubkey-chain ssh pubkey-chain server 10.7.8.9 key-hash sha256 f1:22:49:47:b6:76:74:b2:db:26:fb:13:65:d8: 99:19:e7:9e:24:46:59:be:13:7f:25:27:70:9b: 0e:d2:86:12</pre>	<p>The ASA stores the SSH host key for each SCP server to which it connects. You can manually add or delete servers and their keys from the ASA database if desired.</p> <p>For each server, you can specify the key-string (public key) or key-hash (hashed value) of the SSH host.</p> <p>The <i>key_string</i> is the Base64 encoded RSA public key of the remote peer. You can obtain the public key value from an open SSH client; that is, from the <code>.ssh/id_rsa.pub</code> file. After you submit the Base64 encoded public key, that key is then hashed via SHA-256.</p> <p>The key-hash {md5 sha256} fingerprint enters the already hashed key (using an MD5 or SHA-256 key); for example, a key that you copied from show command output.</p>	

Examples

The following example adds an already hashed host key for the server at 10.86.94.170:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.86.94.170
ciscoasa(config-ssh-pubkey-server)# key-hash sha256
65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2d:bf:a9:2b:85:2e:19
```

The following example adds a host string key for the server at 10.7.8.9:

```
ciscoasa(config)# ssh pubkey-chain
ciscoasa(config-ssh-pubkey-chain)# server 10.7.8.9
ciscoasa(config-ssh-pubkey-server)# key-string
Enter the base 64 encoded RSA public key.
End with the word "exit" on a line by itself
ciscoasa(config-ssh-pubkey-server-string)# c1:b1:30:29:d7:b8:de:6c:97:77:10:d7:46:41:63:87
ciscoasa(config-ssh-pubkey-server-string)# exit
```

Configuring the ASA TFTP Client Path

TFTP is a simple client/server file transfer protocol, which is described in RFC 783 and RFC 1350 Rev. 2. You can configure the ASA as a TFTP *client* so that it can copy files to or from a TFTP *server* (see the “Copying a File to the ASA” section on page 42-17 and “Backing Up Configurations or Other Files” section on page 42-25). In this way, you can back up and propagate configuration files to multiple ASAs.

This section lets you pre-define the path to a TFTP server so you do not need to enter it in commands such as **copy** and **configure net**.

Detailed Steps

Command	Purpose
<pre>tftp-server interface_name server_ip filename</pre> <p>Example:</p> <pre>ciscoasa(config)# tftp-server inside 10.1.4.7 files/config1.cfg ciscoasa(config)# copy tftp: test.cfg</pre> <p>Address or name of remote host [10.1.4.7]?</p> <p>Source filename [files/config1.cfg]?config2.cfg</p> <p>Destination filename [test.cfg]?</p> <p>Accessing tftp://10.1.4.7/files/config2.cfg;int=outside...</p>	<p>Pre-defines the TFTP server address and filename for use with configure net and copy commands. You can override the filename when you enter the command; for example, when you use the copy command, you can take advantage of the pre-defined TFTP server address but still enter any filename at the interactive prompts.</p> <p>For the copy command, enter tftp: to use the tftp-server value instead of tftp://url.</p>

Copying a File to the ASA

This section describes how to copy the application image, ASDM software, a configuration file, or any other file that needs to be downloaded to internal or external flash memory from a TFTP, FTP, SMB, HTTP, HTTPS, or SCP server.

Guidelines

- For the IPS SSP software module, before you download the IPS software to disk0, make sure at least 50% of the flash memory is free. When you install IPS, IPS reserves 50% of the internal flash memory for its file system.
- You cannot have two files with the same name but with different letter case in the same directory in flash memory. For example, if you attempt to download the file, Config.cfg, to a location that contains the file, config.cfg, you receive the following error message:

```
%Error opening disk0:/Config.cfg (File exists).
```

- For information about installing the Cisco SSL VPN client, see the *Cisco AnyConnect VPN Client Administrator Guide*. For information about installing Cisco Secure Desktop on the ASA, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5500 Series Administrators*.
- To configure the ASA to use a specific application image or ASDM image if you have more than one installed, or have installed them in external flash memory, see the [“Configuring the Images and Startup Configuration to Use”](#) section on page 42-21.
- For multiple context mode, you must be in the system execution space.

Detailed Steps

Command	Purpose
<pre>copy [/noconfirm] tftp://server[/path]/src_filename {disk0 disk1}:[/path/]dest_filename</pre> <p>Example: ciscoasa# copy tftp://10.1.1.67/files/context1.cfg disk0:/context1.cfg</p> <p>Address or name of remote host [10.1.1.67]? Source filename [files/context1.cfg]? Destination filename [context1.cfg]? Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b !!!!!!!!!!!! 11143 bytes copied in 5.710 secs (2228 bytes/sec)</p>	Copies from a TFTP server.
<pre>copy [/noconfirm] ftp://[user[:password]@]server[/path]/src_filename {disk0 disk1}:[/path/]dest_filename</pre> <p>Example: ciscoasa# copy ftp://jcrichton:aeryn@10.1.1.67/files/context1.cfg disk0:/contexts/context1.cfg</p> <p>Address or name of remote host [10.1.1.67]? Source username [jcrichton]? Source password [aeryn]? Source filename [files/context1.cfg]? Destination filename [contexts/context1.cfg]? Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b !!!!!!!!!!!! 11143 bytes copied in 5.710 secs (2228 bytes/sec)</p>	Copies from an FTP server.
<pre>copy [/noconfirm] http[s]://[user[:password]@]server[:port][/path]/src_filename {disk0 disk1}:[/path/]dest_filename</pre> <p>Example: ciscoasa# copy https://asun:john@10.1.1.67/files/moya.cfg disk0:/contexts/moya.cfg</p> <p>Address or name of remote host [10.1.1.67]? Source username [asun]? Source password [john]? Source filename [files/moya.cfg]? Destination filename [contexts/moya.cfg]? Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b !!!!!!!!!!!! 11143 bytes copied in 5.710 secs (2228 bytes/sec)</p>	Copies from an HTTP(S) server.

Command	Purpose
<pre>copy [/noconfirm] smb://[user[:password]@]server[/path]/src_filename {disk0 disk1}:[/path/]dest_filename</pre> <p>Example:</p> <pre>ciscoasa# copy /noconfirm smb://chiana:dargo@10.1.1.67/test.xml disk0:/test.xml</pre> <pre>Cryptochecksum: db8ba196 9ad189a8 7f5f501f 1bec469b !!!!!!!!!!!! 11143 bytes copied in 5.710 secs (2228 bytes/sec)</pre>	Copies from an SMB server.
<pre>copy [/noconfirm] scp://[user[:password]@]server[/path]/src_filename[;int=interface_name] {disk0 disk1}:[/path/]dest_filename</pre> <p>Example:</p> <pre>ciscoasa# copy scp://pilot@10.86.94.170/test.cfg disk0:/test.cfg</pre> <pre>Address or name of remote host [10.86.94.170]? Source username [pilot]? Destination filename [test.cfg]? The authenticity of host '10.86.94.170 (10.86.94.170)' can't be established. RSA key fingerprint is <65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d: 2d:bf:a9:2b:85:2e:19> (SHA256) . Are you sure you want to continue connecting (yes/no)? yes Please use the following commands to add the hash key to the configuration: ssh pubkey-chain server 10.86.94.170 key-hash sha256 65:d9:9d:fe:1a:bc:61:aa:64:9d:fc:ee:99:87:38:df:a8:8e:d9:e9:ff:42:de:e8:8d:2 d:bf:a9:2b:85:2e:19 Password: <type in password> !!!!!!!! 6006 bytes copied in 8.160 secs (750 bytes/sec)</pre>	Copies from a SCP server. The int=interface option bypasses the route lookup and always uses the specified interface to reach the SCP server.

Copying a File to the Startup or Running Configuration

You can download a text file to the running or startup configuration from a TFTP, FTP, SMB, HTTP(S) or SCP server, or from the flash memory.

To configure the ASA to use a specific configuration as the startup configuration, see the [“Configuring the File to Boot as the Startup Configuration”](#) section on page 42-22.

Guidelines

When you copy a configuration to the running configuration, you merge the two configurations. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

Detailed Steps

To copy a file to the startup configuration or running configuration, enter one of the following commands for the appropriate download server:

Command	Purpose
<pre>copy [/noconfirm] tftp://server[/path]/src_filename {startup-config running-config}</pre> <p>Example: ciscoasa# copy tftp://10.1.1.67/files/old-running.cfg running-config</p>	Copies from a TFTP server.
<pre>copy [/noconfirm] ftp://[user[:password]@]server[/path]/src_filename {startup-config running-config}</pre> <p>Example: ciscoasa# copy ftp://jcrichon:aeryn@10.1.1.67/files/old-startup.cfg startup-config</p>	Copies from an FTP server.
<pre>copy [/noconfirm] http[s]://[user[:password]@]server[:port][/path]/src_filename {startup-config running-config}</pre> <p>Example: ciscoasa# copy https://asun:john@10.1.1.67/files/new-running.cfg running-config</p>	Copies from an HTTP(S) server.
<pre>copy [/noconfirm] smb://[user[:password]@]server[/path]/src_filename {startup-config running-config}</pre> <p>Example: ciscoasa# copy /noconfirm smb://chiana:dargo@10.1.1.67/new-running.cfg running-config</p>	Copies from an SMB server.
<pre>copy [/noconfirm] scp://[user[:password]@]server[/path]/src_filename[;int=interface_name] {startup-config running-config}</pre> <p>Example: ciscoasa# copy scp://pilot:moya@10.86.94.170/new-startup.cfg startup-config</p>	Copies from a SCP server. The ;int=interface option bypasses the route lookup and always uses the specified interface to reach the SCP server.

Examples

For example, to copy the configuration from a TFTP server, enter the following command:

```
ciscoasa# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
ciscoasa# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
ciscoasa# copy http://209.165.200.228/configs/startup.cfg startup-config
```

Configuring the Images and Startup Configuration to Use

By default, the ASA boots the first application image that it finds in internal flash memory. It also boots the first ASDM image it finds in internal flash memory, or if one does not exist in this location, then in external flash memory. If you have more than one image, you should specify the image that you want to boot. For the ASDM image, if you do not specify the image to boot, even if you have only one image installed, then the ASA inserts the **asdm image** command into the running configuration. To avoid problems with Auto Update (if configured), and to avoid the image search at each startup, you should specify the ASDM image that you want to boot in the startup configuration.

- [Configuring the ASA and ASDM Images to Use, page 42-21](#)
- [Configuring the File to Boot as the Startup Configuration, page 42-22](#)

Configuring the ASA and ASDM Images to Use

To configure the application image to boot, enter the following command:

```
ciscoasa(config)# boot system url
```

where *url* can be one of the following:

- `{disk0:/ | disk1:/}[path/]filename`
- `tftp://[user[:password]@]server[:port]/[path/]filename`



Note The TFTP option is not supported on all models.

You can enter up to four **boot system** command entries to specify different images to boot from in order; the ASA boots the first image it finds successfully. When you enter the **boot system** command, it adds an entry at the bottom of the list. To reorder the boot entries, you must remove all entries using the **clear configure boot system** command, and re-enter them in the order you desire. Only one **boot system tftp** command can be configured, and it must be the first one configured.



Note If the ASA is stuck in a cycle of constant booting, you can reboot the ASA into ROMMON mode. For more information about the ROMMON mode, see the [“Viewing Debugging Messages” section on page 43-1](#).

To configure the ASDM image to boot, enter the following command:

```
ciscoasa(config)# asdm image {disk0:/ | disk1:/}[path/]filename
```

Configuring the File to Boot as the Startup Configuration

By default, the ASA boots from a startup configuration that is a hidden file. You can alternatively set any configuration to be the startup configuration by entering the following command:

```
ciscoasa(config)# boot config {disk0:/ | disk1:/}[path/]filename
```

Using the ROM Monitor to Load an Image

- [Using ROM Monitor for the ASA 5500 Series, page 42-22](#)
- [Using the ROM Monitor for the ASASM, page 42-23](#)

Using ROM Monitor for the ASA 5500 Series

To load a software image to an ASA from the ROM monitor mode using TFTP, perform the following steps:

-
- Step 1** Connect to the ASA console port according to the instructions in the [“Accessing the Appliance Command-Line Interface”](#) section on page 3-1.
- Step 2** Power off the ASA, then power it on.
- Step 3** During startup, press the **Escape** key when you are prompted to enter ROMMON mode.
- Step 4** In ROMMON mode, define the interface settings to the ASA, including the IP address, TFTP server address, gateway address, software image file, and port, as follows:

```
rommon #1> ADDRESS=10.132.44.177
rommon #2> SERVER=10.129.0.30
rommon #3> GATEWAY=10.132.44.1
rommon #4> IMAGE=f1/asa800-232-k8.bin
rommon #5> PORT=Ethernet0/0
Ethernet0/0
Link is UP
MAC Address: 0012.d949.15b8
```



Note Be sure that the connection to the network already exists.

- Step 5** To validate your settings, enter the **set** command.

```
rommon #6> set
ROMMON Variable Settings:
  ADDRESS=10.132.44.177
  SERVER=10.129.0.30
  GATEWAY=10.132.44.1
  PORT=Ethernet0/0
  VLAN=untagged
  IMAGE=f1/asa840-232-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20
```

- Step 6** Ping the TFTP server by entering the **ping server** command.

```
rommon #7> ping server
Sending 20, 100-byte ICMP Echoes to server 10.129.0.30, timeout is 4 seconds:

Success rate is 100 percent (20/20)
```

Step 7 Load the software image by entering the **tftp** command.

```
rommon #8> tftp
ROMMON Variable Settings:
  ADDRESS=10.132.44.177
  SERVER=10.129.0.30
  GATEWAY=10.132.44.1
  PORT=Ethernet0/0
  VLAN=untagged
  IMAGE=f1/asa840-232-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20

tftp f1/asa840-232-k8.bin@10.129.0.30 via 10.132.44.1

Received 14450688 bytes

Launching TFTP Image...
Cisco ASA Security Appliance admin loader (3.0) #0: Mon Mar 5 16:00:07 MST 2011

Loading...N
After the software image is successfully loaded, the ASA automatically exits ROMMON mode.
```

Step 8 To verify that the correct software image has been loaded into the ASA, check the version in the ASA by entering the following command:

```
hostname# show version
```

Using the ROM Monitor for the ASASM

To load a software image to an ASASM from the ROM monitor mode using TFTP, perform the following steps:

- Step 1** Connect to the ASA console port according to the instructions in the [“Accessing the ASA Services Module Command-Line Interface”](#) section on page 3-2.
- Step 2** Make sure that you reload the ASASM image.
- Step 3** During startup, press the **Escape** key when you are prompted to enter ROMMON mode.
- Step 4** In ROMMOM mode, define the interface settings to the ASASM, including the IP address, TFTP server address, gateway address, software image file, port, and VLAN, as follows:

```
rommon #1> ADDRESS=172.16.145.149
rommon #2> SERVER=172.16.171.125
rommon #3> GATEWAY=172.16.145.129
rommon #4> IMAGE=f1/asa851-smp-k8.bin
rommon #5> PORT=Data0
rommon #6> VLAN=1
Data0
Link is UP
MAC Address: 0012.d949.15b8
```



Note Be sure that the connection to the network already exists.

Step 5 To validate your settings, enter the **set** command.

```
rommon #7> set
ROMMON Variable Settings:
  ADDRESS=172.16.145.149
  SERVER=172.16.171.125
  GATEWAY=172.16.145.129
  PORT=Data0
  VLAN=1
  IMAGE=f1/asa851-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=2
  RETRY=20
```

Step 6 Ping the TFTP server by entering the **ping server** command.

```
rommon #8> ping server
Sending 20, 100-byte ICMP Echoes to server 172.16.171.125, timeout is 2 seconds:

Success rate is 100 percent (20/20)
```

Step 7 Load the software image by entering the **tftp** command.

```
rommon #9> tftp
Clearing EOBC receive queue ...
cmostime_set = 1
ROMMON Variable Settings:
  ADDRESS=172.16.145.149
  SERVER=172.16.171.125
  GATEWAY=172.16.145.129
  PORT=Data0
  VLAN=1
  IMAGE=f1/asa851-smp-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=2
  RETRY=20

tftp f1/asa851-smp-k8.bin@172.16.171.125 via 172.16.145.129
Starting download. Press ESC to abort.
```

After the software image is successfully loaded, the ASASM automatically exits ROMMON mode.



Note You must download the image to the system flash separately after ROMMON boot is complete; booting the module into ROMMON mode does not preserve the system image across reloads.

Step 8 To verify that the correct software image has been loaded into the ASASM, check the version by entering the following command:

```
hostname# show version
```

Backing Up Configurations or Other Files

- [Backing up the Single Mode Configuration or Multiple Mode System Configuration, page 42-25](#)
- [Backing Up a Context Configuration or Other File in Flash Memory, page 42-26](#)
- [Backing Up a Context Configuration within a Context, page 42-27](#)
- [Copying the Configuration from the Terminal Display, page 42-27](#)
- [Backing Up Additional Files Using the Export and Import Commands, page 42-27](#)
- [Using a Script to Back Up and Restore Files, page 42-28](#)

Backing up the Single Mode Configuration or Multiple Mode System Configuration

In single context mode or from the system configuration in multiple mode, you can copy the startup configuration or running configuration to an external server or to the local flash memory.

Detailed Steps

Command	Purpose
<pre>copy [/noconfirm] {startup-config running-config} tftp://server[/path]/dst_filename</pre> <p>Example: ciscoasa# copy running-config tftp://10.1.1.67/files/new-running.cfg</p>	Copies to a TFTP server.
<pre>copy [/noconfirm] {startup-config running-config} ftp://[user[:password]@]server[/path]/dst_filename</pre> <p>Example: ciscoasa# copy startup-config ftp://jcrichton:aeryn@10.1.1.67/files/new-startup.cfg</p>	Copies to an FTP server.
<pre>copy [/noconfirm] {startup-config running-config} smb://[user[:password]@]server[/path]/dst_filename</pre> <p>Example: ciscoasa# copy /noconfirm running-config smb://chiana:dargo@10.1.1.67/new-running.cfg</p>	Copies to an SMB server.

Command	Purpose
<pre>copy [/noconfirm] {startup-config running-config} scp://[user[:password]@]server[/path]/dst_filename[;int=interface_name]</pre> <p>Example:</p> <pre>ciscoasa# copy startup-config scp://pilot:moya@10.86.94.170/new-startup.cfg</pre>	Copies to a SCP server. The int=interface option bypasses the route lookup and always uses the specified interface to reach the SCP server.
<pre>copy [/noconfirm] {startup-config running-config} {disk0 disk1}:[path/]dst_filename</pre> <p>Example:</p> <pre>ciscoasa# copy /noconfirm running-config disk0:/new-running.cfg</pre>	Copies to the local flash memory. Be sure that the destination directory exists. If it does not exist, first create the directory using the mkdir command.

Backing Up a Context Configuration or Other File in Flash Memory

Copy context configurations or other files that are on the local flash memory by entering one of the following commands in the system execution space.

Detailed Steps

Command	Purpose
<pre>copy [/noconfirm] {disk0 disk1}:[path/]src_filename tftp://server[/path]/dst_filename</pre> <p>Example:</p> <pre>ciscoasa# copy disk0:/asa-os.bin tftp://10.1.1.67/files/asa-os.bin</pre>	Copies from flash to a TFTP server.
<pre>copy [/noconfirm] {disk0 disk1}:[path/]src_filename ftp://[user[:password]@]server[/path]/dst_filename</pre> <p>Example:</p> <pre>ciscoasa# copy disk0:/asa-os.bin ftp://jcrichon:aeryn@10.1.1.67/files/asa-os.bin</pre>	Copies from flash to an FTP server.
<pre>copy [/noconfirm] {disk0 disk1}:[path/]src_filename smb://[user[:password]@]server[/path]/dst_filename</pre> <p>Example:</p> <pre>ciscoasa# copy /noconfirm copy disk0:/asdm.bin smb://chiana:dargo@10.1.1.67/asdm.bin</pre>	Copies from flash to an SMB server.

Command	Purpose
<pre>copy [/noconfirm] {disk0 disk1}:[path/]src_filename scp://[user[:password]@]server[/path]/dst_filename[;int=interface_name]</pre> <p>Example:</p> <pre>ciscoasa# copy disk0:/context1.cfg scp://pilot:moya@10.86.94.170/context1.cfg</pre>	<p>Copies from flash to SCP server. The ;int=interface option bypasses the route lookup and always uses the specified interface to reach the SCP server.</p>
<pre>copy [/noconfirm] {disk0 disk1}:[path/]src_filename {disk0 disk1}:[path/]dst_filename</pre> <p>Example:</p> <pre>ciscoasa# copy /noconfirm disk1:/file1.cfg disk0:/file1.cfgnew-running.cfg</pre>	<p>Copies from flash to the local flash memory. Be sure that the destination directory exists. If it does not exist, first create the directory using the mkdir command.</p>

Backing Up a Context Configuration within a Context

In multiple context mode, from within a context, you can perform the following backups:

- To copy the running configuration to the startup configuration server (connected to the admin context), enter the following command:

```
ciscoasa/contexta# copy running-config startup-config
```

- To copy the running configuration to a TFTP server connected to the context network, enter the following command:

```
ciscoasa/contexta# copy running-config tftp:/server[/path]/filename
```

Copying the Configuration from the Terminal Display

To print the configuration to the terminal, enter the following command:

```
ciscoasa# show running-config
```

Copy the output from this command, and then paste the configuration into a text file.

Backing Up Additional Files Using the Export and Import Commands

Additional files essential to your configuration might include the following:

- Files that you import using the **import webvpn** command. Currently, these files include customizations, URL lists, web content, plug-ins, and language translations.
- DAP policies (dap.xml).
- CSD configurations (data.xml).
- Digital keys and certificates.
- Local CA user database and certificate status files.

The CLI lets you back up and restore individual elements of your configuration using the **export** and **import** commands.

To back up these files, for example, those files that you imported with the **import webvpn** command or certificates, perform the following steps:

Step 1 Run the applicable **show** command(s) as follows:

```
ciscoasa # show import webvpn plug-in
ica
rdp
ssh, telnet
vnc
```

Step 2 Run the **export** command for the file that you want to back up (in this example, the rdp file):

```
ciscoasa # export webvpn plug-in protocol rdp tftp://tftpserver/backupfilename
```

Using a Script to Back Up and Restore Files

You can use a script to back up and restore the configuration files on your ASA, including all extensions that you import via the **import webvpn** CLI, the CSD configuration XML files, and the DAP configuration XML file. For security reasons, we do not recommend that you perform automated backups of digital keys and certificates or the local CA key.

This section provides instructions for doing so and includes a sample script that you can use as is or modify as your environment requires. The sample script is specific to a Linux system. To use it for a Microsoft Windows system, you need to modify it using the logic of the sample.



Note

The existing CLI lets you back up and restore individual files using the **copy**, **export**, and **import** commands. It does not, however, have a facility that lets you back up all ASA configuration files in one operation. Running the script facilitates the use of multiple CLIs.

This section includes the following topics:

- [Prerequisites, page 42-28](#)
- [Running the Script, page 42-29](#)
- [Sample Script, page 42-29](#)

Prerequisites

To use a script to back up and restore an ASA configuration, first perform the following tasks:

- Install Perl with an Expect module.
- Install an SSH client that can reach the ASA.
- Install a TFTP server to send files from the ASA to the backup site.

Another option is to use a commercially available tool. You can put the logic of this script into such a tool.

Running the Script

To run a backup-and-restore script, perform the following steps:

-
- Step 1** Download or cut-and-paste the script file to any location on your system.
 - Step 2** At the command line, enter **Perl** *scriptname*, where *scriptname* is the name of the script file.
 - Step 3** Press **Enter**.
 - Step 4** The system prompts you for values for each option. Alternatively, you can enter values for the options when you enter the **Perl** *scriptname* command before you press **Enter**. Either way, the script requires that you enter a value for each option.
 - Step 5** The script starts running, printing out the commands that it issues, which provides you with a record of the CLIs. You can use these CLIs for a later restore, which is particularly useful if you want to restore only one or two files.
-

Sample Script

```
#!/usr/bin/perl
#Function: Backup/restore configuration/extensions to/from a TFTP server.
#Description: The objective of this script is to show how to back up
configurations/extensions before the backup/restore command is developed.
# It currently backs up the running configuration, all extensions imported via "import
webvpn" command, the CSD configuration XML file, and the DAP configuration XML file.
#Requirements: Perl with Expect, SSH to the ASA, and a TFTP server.
#Usage: backupasa -option option_value
#       -h: ASA hostname or IP address
#       -u: User name to log in via SSH
#       -w: Password to log in via SSH
#       -e: The Enable password on the security appliance
#       -p: Global configuration mode prompt
#       -s: Host name or IP address of the TFTP server to store the configurations
#       -r: Restore with an argument that specifies the file name. This file is produced
during backup.
#If you don't enter an option, the script will prompt for it prior to backup.
#
#Make sure that you can SSH to the ASA.

use Expect;
use Getopt::Std;

#global variables
%options=();
$restore = 0; #does backup by default
$restore_file = '';
$asa = '';
$storage = '';
$user = '';
$password = '';
$enable = '';
$prompt = '';
$date = `date +%F`;
chop($date);
my $exp = new Expect();

getopts("h:u:p:w:e:s:r:", \%options);
do process_options();
```

```

do login($exp);
do enable($exp);
if ($restore) {
    do restore($exp,$restore_file);
}
else {
    $restore_file = "$prompt-restore-$date.cli";
    open(OUT,">$restore_file") or die "Can't open $restore_file\n";
    do running_config($exp);
    do lang_trans($exp);
    do customization($exp);
    do plugin($exp);
    do url_list($exp);
    do webcontent($exp);
    do dap($exp);
    do csd($exp);
    close(OUT);
}
do finish($exp);

sub enable {
    $obj = shift;
    $obj->send("enable\n");
    unless ($obj->expect(15, 'Password:')) {
        print "timed out waiting for Password:\n";
    }
    $obj->send("$enable\n");
    unless ($obj->expect(15, "$prompt#")) {
        print "timed out waiting for $prompt#\n";
    }
}

sub lang_trans {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn translation-table\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        s/^\s+//;
        s/\s+$//;
        next if /show import/ or /Translation Tables/;
        next unless (/^\.+s+.$/);
        ($lang, $transtable) = split(/\s+/, $_);
        $cli = "export webvpn translation-table $transtable language $lang
$storage/$prompt-$date-$transtable-$lang.po";
        $ocli = $cli;
        $ocli =~ s/^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

sub running_config {
    $obj = shift;
    $obj->clear_accum();
    $cli = "copy /noconfirm running-config $storage/$prompt-$date.cfg";
    print "$cli\n";
    $obj->send("$cli\n");
}

```

```

$obj->expect(15, "$prompt#" );
}

sub customization {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn customization\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /\s*$/;
        $cli = "export webvpn customization $_ $storage/$prompt-$date-cust-$_.xml";
        $ocli = $cli;
        $ocli =~ s/^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

sub plugin {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn plug-in\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /\s*$/;
        $cli = "export webvpn plug-in protocol $_ $storage/$prompt-$date-plugin-$_.jar";
        $ocli = $cli;
        $ocli =~ s/^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

sub url_list {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn url-list\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /\s*$/ or /No bookmarks/;
        $cli="export webvpn url-list $_ $storage/$prompt-$date-urllist-$_.xml";
        $ocli = $cli;
        $ocli =~ s/^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

```

```

    }
}

sub dap {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("dir dap.xml\n");
    $obj->expect(15, "$prompt#" );

    $output = $obj->before();
    return 0 if($output =~ /Error/);

    $cli="copy /noconfirm dap.xml $storage/$prompt-$date-dap.xml";
    $ocli="copy /noconfirm $storage/$prompt-$date-dap.xml disk0:/dap.xml";
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

sub csd {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("dir sdesktop\n");
    $obj->expect(15, "$prompt#" );

    $output = $obj->before();
    return 0 if($output =~ /Error/);

    $cli="copy /noconfirm sdesktop/data.xml $storage/$prompt-$date-data.xml";
    $ocli="copy /noconfirm $storage/$prompt-$date-data.xml disk0:/sdesktop/data.xml";
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

sub webcontent {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn webcontent\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        s/^\s+//;
        s/\s+$//;
        next if /show import/ or /No custom/;
        next unless (/^.\s+.$/);
        ($url, $type) = split(/\s+/, $_);
        $turl = $url;
        $turl =~ s/\/\+//;
        $turl =~ s/\/+\/-//;
        $cli = "export webvpn webcontent $url $storage/$prompt-$date-$turl";
        $ocli = $cli;
        $ocli =~ s/^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

```



```

sub login {
    $obj = shift;
    $obj->raw_pty(1);
    $obj->log_stdout(0); #turn off console logging.
    $obj->spawn("/usr/bin/ssh $user@$asa") or die "can't spawn ssh\n";
    unless ($obj->expect(15, "password:" )) {
        die "timeout waiting for password:\n";
    }

    $obj->send("$password\n");

    unless ($obj->expect(15, "$prompt>" )) {
        die "timeout waiting for $prompt>\n";
    }
}

sub finish {
    $obj = shift;
    $obj->hard_close();
    print "\n\n";
}

sub restore {
    $obj = shift;
    my $file = shift;
    my $output;
    open(IN,$file) or die "can't open $file\n";
    while (<IN>) {
        $obj->send("$_");
        $obj->expect(15, "$prompt#" );
        $output = $obj->before();
        print "$output\n";
    }
    close(IN);
}

sub process_options {
    if (defined($options{s})) {
        $tstr= $options{s};
        $storage = "tftp://$tstr";
    }
    else {
        print "Enter TFTP host name or IP address:";
        chop($tstr=<>);
        $storage = "tftp://$tstr";
    }
    if (defined($options{h})) {
        $asa = $options{h};
    }
    else {
        print "Enter ASA host name or IP address:";
        chop($asa=<>);
    }

    if (defined ($options{u})) {
        $user= $options{u};
    }
    else {
        print "Enter user name:";
        chop($user=<>);
    }

    if (defined ($options{w})) {

```

```

    $password= $options{w};
}
else {
    print "Enter password:";
    chop($password=<>);
}
if (defined ($options{p})) {
    $prompt= $options{p};
}
else {
    print "Enter ASA prompt:";
    chop($prompt=<>);
}
if (defined ($options{e})) {
    $enable = $options{e};
}
else {
    print "Enter enable password:";
    chop($enable=<>);
}

if (defined ($options{r})) {
    $restore = 1;
    $restore_file = $options{r};
}
}

```

Downgrading Your Software

When you upgrade to Version 8.3, your configuration is migrated. The old configuration is automatically stored in flash memory. For example, when you upgrade from Version 8.2(1) to 8.3(1), the old 8.2(1) configuration is stored in flash memory in a file called 8_2_1_0_startup_cfg.sav.



Note

You must manually restore the old configuration before downgrading.

This section describes how to downgrade and includes the following topics:

- [Information About Activation Key Compatibility, page 42-34](#)
- [Performing the Downgrade, page 42-35](#)

Information About Activation Key Compatibility

Your activation key remains compatible if you upgrade to the latest version from any previous version. However, you might have issues if you want to maintain downgrade capability:

- Downgrading to Version 8.1 or earlier versions—After you upgrade, if you activate additional feature licenses that were introduced *before 8.2*, the activation key continues to be compatible with earlier versions if you downgrade. However if you activate feature licenses that were introduced in Version 8.2 or later versions, the activation key is not backwards compatible. If you have an incompatible license key, see the following guidelines:
 - If you previously entered an activation key in an earlier version, the ASA uses that key (without any of the new licenses you activated in Version 8.2 or later versions).

- If you have a new system and do not have an earlier activation key, you need to request a new activation key compatible with the earlier version.
- Downgrading to Version 8.2 or earlier versions—Version 8.3 introduced more robust time-based key usage as well as failover license changes:
 - If you have more than one time-based activation key active, when you downgrade, only the most recently activated time-based key can be active. Any other keys are made inactive.
 - If you have mismatched licenses on a failover pair, downgrading will disable failover. Even if the keys are matching, the license used will no longer be a combined license.

Performing the Downgrade

To downgrade from Version 8.3, perform the following steps:

Detailed Steps

Step 1 Enter the following command:

```
ciscoasa(config)# downgrade [/noconfirm] old_image_url old_config_url [activation-key
old_key]
```

Where the **/noconfirm** option downgrades without prompting. The *image_url* is the path to the old image on disk0, disk1, tftp, ftp, or smb. The *old_config_url* is the path to the saved, premigration configuration (by default, this configuration was saved on disk0). If you need to revert to a pre-8.3 activation key, you can enter the old activation key.

This command is a shortcut for completing the following functions:

1. Clearing the boot image configuration (**clear configure boot**).
2. Setting the boot image to be the old image (**boot system**).
3. (Optional) Entering a new activation key (**activation-key**).
4. Saving the running configuration to startup (**write memory**). This action sets the BOOT environment variable to the old image, so when you reload, the old image is loaded.
5. Copying the old configuration to the startup configuration (**copy *old_config_url* startup-config**).
6. Reloading (**reload**).

For example:

```
ciscoasa(config)# downgrade /noconfirm disk0:/asa821-k8.bin disk0:/8_2_1_0_startup_cfg.sav
```

Configuring Auto Update

This section includes the following topics:

- [Information About Auto Update, page 42-36](#)
- [Guidelines and Limitations, page 42-39](#)
- [Configuring Communication with an Auto Update Server, page 42-39](#)
- [Configuring Client Updates as an Auto Update Server, page 42-41](#)

- [Viewing Auto Update Status, page 42-42](#)

Information About Auto Update

Auto Update is a protocol specification that allows an Auto Update Server to download configurations and software images to many ASAs and can provide basic monitoring of the ASAs from a central location.

- [Auto Update Client or Server, page 42-36](#)
- [Auto Update Benefits, page 42-36](#)
- [Auto Update Server Support in Failover Configurations, page 42-36](#)

Auto Update Client or Server

The ASA can be configured as either a client or a server. As an Auto Update client, it periodically polls the Auto Update Server for updates to software images and configuration files. As an Auto Update Server, it issues updates for ASAs configured as Auto Update clients.

Auto Update Benefits

Auto Update is useful in solving many issues facing administrators for ASA management, such as:

- Overcoming dynamic addressing and NAT challenges.
- Committing configuration changes in one action.
- Providing a reliable method for updating software.
- Leveraging well-understood methods for high availability (failover).
- Providing flexibility with an open interface.
- Simplifying security solutions for Service Provider environments.

The Auto Update specification provides the infrastructure necessary for remote management applications to download ASA configurations, software images, and to perform basic monitoring from a centralized location or multiple locations.

The Auto Update specification allows the Auto Update server to either push configuration information and send requests for information to the ASA, or to pull configuration information by having the ASA periodically poll the Auto Update server. The Auto Update server can also send a command to the ASA to send an immediate polling request at any time. Communication between the Auto Update server and the ASA requires a communications path and local CLI configuration on each ASA.

Auto Update Server Support in Failover Configurations

You can use the Auto Update Server to deploy software images and configuration files to ASAs in an Active/Standby failover configuration. To enable Auto Update on an Active/Standby failover configuration, enter the Auto Update Server configuration on the primary unit in the failover pair.

The following restrictions and behaviors apply to Auto Update Server support in failover configurations:

- Only single mode, Active/Standby configurations are supported.
- When loading a new platform software image, the failover pair stops passing traffic.

- When using LAN-based failover, new configurations must not change the failover link configuration. If they do, communication between the units will fail.
- Only the primary unit will perform the call home to the Auto Update Server. The primary unit must be in the active state to call home. If it is not, the ASA automatically fails over to the primary unit.
- Only the primary unit downloads the software image or configuration file. The software image or configuration is then copied to the secondary unit.
- The interface MAC address and hardware-serial ID is from the primary unit.
- The configuration file stored on the Auto Update Server or HTTP server is for the primary unit only.

Auto Update Process Overview

The following is an overview of the Auto Update process in failover configurations. This process assumes that failover is enabled and operational. The Auto Update process cannot occur if the units are synchronizing configurations, if the standby unit is in the failed state for any reason other than SSM card failure, or if the failover link is down.

1. Both units exchange the platform and ASDM software checksum and version information.
2. The primary unit contacts the Auto Update Server. If the primary unit is not in the active state, the ASA first fails over to the primary unit and then contacts the Auto Update Server.
3. The Auto Update Server replies with software checksum and URL information.
4. If the primary unit determines that the platform image file needs to be updated for either the active or standby unit, the following occurs:
 - a. The primary unit retrieves the appropriate files from the HTTP server using the URL from the Auto Update Server.
 - b. The primary unit copies the image to the standby unit and then updates the image on itself.
 - c. If both units have new image, the secondary (standby) unit is reloaded first.
 - If hitless upgrade can be performed when secondary unit boots, then the secondary unit becomes the active unit and the primary unit reloads. The primary unit becomes the active unit when it has finished loading.
 - If hitless upgrade cannot be performed when the standby unit boots, then both units reload at the same time.
 - d. If only the secondary (standby) unit has new image, then only the secondary unit reloads. The primary unit waits until the secondary unit finishes reloading.
 - e. If only the primary (active) unit has new image, the secondary unit becomes the active unit, and the primary unit reloads.
 - f. The update process starts again at Step 1.
5. If the ASA determines that the ASDM file needs to be updated for either the primary or secondary unit, the following occurs:
 - a. The primary unit retrieves the ASDM image file from the HTTP server using the URL provided by the Auto Update Server.
 - b. The primary unit copies the ASDM image to the standby unit, if needed.
 - c. The primary unit updates the ASDM image on itself.
 - d. The update process starts again at Step 1.
6. If the primary unit determines that the configuration needs to be updated, the following occurs:

- a. The primary unit retrieves the configuration file from the using the specified URL.
 - b. The new configuration replaces the old configuration on both units simultaneously.
 - c. The update process begins again at Step 1.
7. If the checksums match for all image and configuration files, no updates are required. The process ends until the next poll time.

Monitoring the Auto Update Process

You can use the **debug auto-update client** or **debug fover cmd-exe** commands to display the actions performed during the Auto Update process. The following is sample output from the **debug auto-update client** command.

```
Auto-update client: Sent DeviceDetails to /cgi-bin/dda.pl of server 192.168.0.21
Auto-update client: Processing UpdateInfo from server 192.168.0.21
Component: asdm, URL: http://192.168.0.21/asdm.bint, checksum:
0x94bced0261cc992ae710faf8d244cf32
Component: config, URL: http://192.168.0.21/config-rms.xml, checksum:
0x67358553572688a805a155af312f6898
Component: image, URL: http://192.168.0.21/cdisk73.bin, checksum:
0x6d091b43ce96243e29a62f2330139419
Auto-update client: need to update img, act: yes, stby yes
name
ciscoasa(config)# Auto-update client: update img on stby unit...
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 9001, len = 1024
auto-update: Fover file copy waiting at clock tick 6129280
fover_parse: Rcvd file copy ack, ret = 0, seq = 4
auto-update: Fover filecopy returns value: 0 at clock tick 6150260, upd time 145980 msecs
Auto-update client: update img on active unit...
fover_parse: Rcvd image info from mate
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
```

```

auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
Beginning configuration replication: Sending to mate.
auto-update: HA safe reload: reload active waiting with mate state: 50
auto-update: HA safe reload: reload active waiting with mate state: 50

auto-update: HA safe reload: reload active waiting with mate state: 80
      Sauto-update: HA safe reload: reload active unit at clock tick: 6266860
Auto-update client: Succeeded: Image, version: 0x6d091b43ce96243e29a62f2330139419

```

The following syslog message is generated if the Auto Update process fails:

```
%ASA4-612002: Auto Update failed: file version: version reason: reason
```

The *file* is “image”, “asdm”, or “configuration”, depending on which update failed. The *version* is the version number of the update. And the *reason* is the reason that the update failed.

Guidelines and Limitations

- If HTTPS is chosen as the protocol to communicate with the Auto Update server, the ASA uses SSL, which requires the ASA to have a DES or 3DES license.
- Auto Update is supported in single context mode only.

Configuring Communication with an Auto Update Server

Detailed Steps

To configure the ASA as an Auto Update client, perform the following steps:

- Step 1** To specify the URL of the Auto Update Server, enter the following command:

```
ciscoasa(config)# auto-update server url [source interface] [verify-certificate]
```

where *url* has the following syntax:

```
http[s]://[user:password@]server_ip[:port]/pathname
```

SSL is used when **https** is specified. The *user* and *password* arguments of the URL are used for basic authentication when logging in to the server. If you use the **write terminal**, **show configuration** or **show tech-support** commands to view the configuration, the user and password are replaced with ‘*****’.

The default port is 80 for HTTP and 443 for HTTPS.

The **source interface** keyword and argument specify which interface to use when sending requests to the Auto Update Server. If you specify the same interface specified by the **management-access** command, the Auto Update requests travel over the same IPsec VPN tunnel used for management access.

The **verify-certificate** keyword verifies the certificate returned by the Auto Update Server.

- Step 2** (Optional) To identify the device ID to send when communicating with the Auto Update Server, enter the following command:

```
ciscoasa(config)# auto-update device-id {hardware-serial | hostname | ipaddress [if-name] | mac-address [if-name] | string text}
```

The identifier used is determined by specifying one of the following parameters:

- The *hardware-serial* argument specifies the ASA serial number.
- The *hostname* argument specifies the ASA hostname.
- The **ipaddress** keyword specifies the IP address of the specified interface. If the interface name is not specified, it uses the IP address of the interface used to communicate with the Auto Update Server.
- The **mac-address** keyword specifies the MAC address of the specified interface. If the interface name is not specified, it uses the MAC address of the interface used to communicate with the Auto Update Server.
- The **string** keyword specifies the specified text identifier, which cannot include white space or the characters ‘, “, , >, & and ?.

Step 3 (Optional) To specify how often to poll the Auto Update Server for configuration or image updates, enter the following command:

```
ciscoasa(config)# auto-update poll-period poll-period [retry-count [retry-period]]
```

The *poll-period* argument specifies how often (in minutes) to check for an update. The default is 720 minutes (12 hours).

The *retry-count* argument specifies how many times to try reconnecting to the server if the first attempt fails. The default is zero.

The *retry-period* argument specifies how long to wait (in minutes) between retries. The default is five minutes.

Step 4 (Optional) To schedule a specific time for the ASA to poll the Auto Update Server, enter the following command:

```
ciscoasa(config)# auto-update poll-at days-of-the-week time [randomize minutes]  
[retry_count [retry_period]]
```

The *days-of-the-week* argument is any single day or combination of days: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Other possible values are daily (Monday through Sunday), weekdays (Monday through Friday), and weekends (Saturday and Sunday).

The *time* argument specifies the time in the format HH:MM at which to start the poll. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

The **randomize** *minutes* keyword and argument specify the period to randomize the poll time following the specified start time. The range is from 1 to 1439 minutes.

The *retry_count* argument specifies how many times to try reconnecting to the Auto Update Server if the first attempt fails. The default is zero.

The *retry_period* argument specifies how long to wait between connection attempts. The default is five minutes. The range is from 1 to 35791 minutes.

Step 5 (Optional) If the Auto Update Server has not been contacted for a certain period of time, entering the following command causes it to stop passing traffic:

```
ciscoasa(config)# auto-update timeout period
```

The *period* argument specifies the timeout period in minutes between 1 and 35791. The default is to never time out (zero minutes). To restore the default, enter the **no** form of this command.

Use the **auto-update timeout** command to be sure that the ASA has the most recent image and configuration. This condition is reported with system log message 201008.

In the following example, an ASA is configured to poll an Auto Update Server with the IP address 209.165.200.224, at port number 1742, from the outside interface, with certificate verification.

The ASA is also configured to use the hostname as the device ID and to poll an Auto Update Server every Friday and Saturday night at a random time between 10:00 p.m. and 11:00 p.m. On a failed polling attempt, the ASA will try to reconnect to the Auto Update Server ten times, and will wait three minutes between attempts at reconnecting, as shown in the following example:

```
ciscoasa(config)# auto-update server
https://jcrichton:farscape@209.165.200.224:1742/management source outside
verify-certificate
ciscoasa (config)# auto-update device-id hostname
hostname (config)# auto-update poll-at Friday Saturday 22:00 randomize 60 2 10
```

Configuring Client Updates as an Auto Update Server

Entering the **client-update** command enables updates for ASAs configured as Auto Update clients and lets you specify the type of software component (ASDM or boot image), the type or family of ASA, revision numbers to which the update applies, and a URL or IP address from which to obtain the update.

To configure the ASA as an Auto Update Server, perform the following steps:

Step 1 To enable client update, enter the following command:

```
ciscoasa(config)# client-update enable
```

Step 2 Configure the following parameters for the **client-update** command that you want to apply to the ASAs:

```
client-update {component {asdm | image} | device-id dev_string |
family family_name | type type} url url-string rev-nums rev-nums}
```

The **component** {**asdm** | **image**} parameter specifies the software component, either ASDM or the boot image of the ASA.

The **device-id** *dev_string* parameter specifies a unique string that the Auto Update client uses to identify itself. The maximum length is 63 characters.

The **family** *family_name* parameter specifies the family name that the Auto Update client uses to identify itself. It can be asa, pix, or a text string with a maximum length of seven characters.

The **rev-nums** *rev-nums* parameter specifies the software or firmware images for this client. Enter up to four, in any order, separated by commas.

The **type** *type* parameter specifies the type of clients to notify of a client update. Because this command is also used to update Windows clients, the list of clients includes several Windows operating systems. The ASAs in the list may include the following:

- asa5505: Cisco 5505 ASA
- asa5510: Cisco 5510 ASA
- asa5520: Cisco 5520 ASA
- asa5540: Cisco 5540 ASA

The **url** *url-string* parameter specifies the URL for the software/firmware image. This URL must point to a file appropriate for this client. For all Auto Update clients, you must use the protocol “http://” or “https://” as the prefix for the URL.

Configure the parameters for the client update that you want to apply to all ASAs of a particular type. That is, specify the type of ASA and the URL or IP address from which to get the updated image. In addition, you must specify a revision number. If the revision number of the remote ASA matches one of the specified revision numbers, there is no need to update the client, and the update is ignored.

To configure a client update for Cisco 5520 ASAs, enter the following command:

```
ciscoasa(config)# client-update type asa5520 component asdm url  
http://192.168.1.114/aus/asdm601.bin rev-nums 8.0(1)
```

Viewing Auto Update Status

To view the Auto Update status, enter the following command:

```
ciscoasa(config)# show auto-update
```

The following is sample output from the **show auto-update** command:

```
ciscoasa(config)# show auto-update  
  
Server: https://*****@209.165.200.224:1742/management.cgi?1276  
Certificate will be verified  
Poll period: 720 minutes, retry count: 2, retry period: 5 minutes  
Timeout: none  
Device ID: host name [corporate]  
Next poll in 4.93 minutes  
Last poll: 11:36:46 PST Tue Nov 13 2004  
Last PDM update: 23:36:46 PST Tue Nov 12 2004
```

Feature History for Software and Configurations

Table 42-2 lists each feature change and the platform release in which it was implemented.

Table 42-2 Feature History for Software and Configurations

Feature Name	Platform Releases	Feature Information
Secure Copy client	9.1(5)	<p>The ASA now supports the Secure Copy (SCP) client to transfer files to and from a SCP server.</p> <p>We introduced the following commands: ssh pubkey-chain, server (ssh pubkey-chain), key-string, key-hash, ssh stricthostkeycheck.</p> <p>We modified the following command: copy scp.</p>